

AMENDMENTS

In the Claims

The following is a marked-up version of the claims with the language that is underlined ("___") being added and the language that contains strikethrough ("—") being deleted:

1. (Currently Amended) A method of monitoring response time of function associated with a software component, comprising:
 - operating on a bytecode representation of a function to be instrumented by inserting an instrumentation code in the bytecode representation of the function without modifying respective source code of the function and while classes of the function are being loaded for execution and incorporating instrumentation hooks into the bytecode representation prior to loading and initialization of a class containing the function by a virtual machine;
 - generating a call to an Application Response Measurement (ARM) agent to cause the ARM agent to effect generation of a start time marker upon start of execution of the function and a stop time marker upon completion of execution of the function, wherein the ARM agent is one of a plurality of agents of an ARM protocol; ~~and~~
generating a correlator, the correlator configured to identify a relationship of the function with at least one other function, the correlator being removed in response to completion of the function and the at least one other function; and
 - utilizing the start and stop time markers to determine a response time of the function.

2. (Canceled)

3. (Previously Presented) The method of claim 1, further comprising:
registering the function with the ARM agent prior to generation of the start and stop time markers.
4. (Previously Presented) The method of claim 1, wherein the instrumentation code causes generation of the start and stop time markers without modifying instructions associated with execution of the function.
5. (Previously Presented) The method of claim 1, wherein the ARM agent generates a record corresponding to the function for storing the response time associated with the function.
6. (Previously Presented) The method of claim 5, wherein the record includes a field for identifying a parent, if any, of the function in a hierarchical parent-child transaction chain.
7. (Previously Presented) The method of claim 6, wherein the record includes another field for identifying a top level transaction in the parent-child transaction chain.
8. (Previously Presented) The method of claim 1, wherein the software component includes at least one of the following: a server page, a servlet of a server side component, a driver, a naming and directory interface (NDI) or remote method invocation (RMI) component.
9. (Previously Presented) The method of claim 8, wherein the function of the software component includes at least one of the following: a service method of a server page, a doFilter, a doGet, a doPost or a service method of a servlet, a getConnection, executeQuery, or selected methods of driver, or remote, local or home interface methods of a server side component.

10. (Canceled)

11. (Previously Presented) The method of claim 1, further comprising:
storing the response time in a database.

12. (Previously Presented) The method of claim 1, further comprising:
displaying the response time to a user.

13. (Currently Amended) A system ~~comprising~~; comprising:

a memory component; and

a processor configured to monitor a response time of a function associated with a software component, the processor configured to implement:

an instrumentation engine for operating on a bytecode representation of a function to be instrumented by inserting instrumentation code in the bytecode representation of the function without modifying respective source code of the function and while classes of the function are being loaded for execution, the instrumentation code effecting generation of a start time marker and a stop time marker upon resumption and completion, respectively, of the function, the instrumentation code further configured to incorporate instrumentation hooks into the bytecode representation prior to loading and initialization of a class containing the function by a virtual machine;

an interface module being invoked by the instrumentation code upon start and completion of the function;

an application response measurement (ARM) agent in communication with the interface module;

a correlator configured to identify a relationship of the function with at least one other function, the correlator being removed in response to completion of the function and the at least one other function;

wherein the interface module, upon invocation by the instrumentation code, calls the ARM agent to cause generation of the start and stop time markers by the ARM agent, and wherein the ARM agent is one of a plurality of agents of an ARM protocol; and

an analysis and presentation module in communication with the ARM agent for presenting the response time to a user and/or storing the response time in a database.

14. (Previously Presented) The system of claim 13, wherein the instrumentation engine inserts the instrumentation code prior to loading of a class containing the function by a virtual machine.

15. (Previously Presented) The system of claim 13, wherein the instrumentation engine inserts the instrumentation code in the bytecode representation without modifying instructions associated with execution of the function.

16. – 21. (Canceled).

22. (Currently Amended) A system of monitoring response time of a function associated with a software component, comprising:

means for operating on a bytecode representation of a function to be instrumented by inserting an instrumentation code in the bytecode representation of the function without modifying respective source code of the function and while classes of the function are being loaded for execution and incorporating instrumentation hooks into the bytecode representation prior to loading and initialization of a class containing the function by a virtual machine;

means for generating a call to an Application Response Measurement (ARM) agent to cause the ARM agent to effect generation of a start time marker upon start of execution of the function and a stop time marker upon completion of execution of the function, wherein the ARM agent is one of a plurality of agents of an ARM protocol; and

means for generating a correlator, the correlator configured to identify a relationship of the function with at least one other function, the correlator being removed in response to completion of the function and the at least one other function; and

means for utilizing the start and stop time markers to determine a response time of the function.

23. (Previously Presented) The method of claim 22, further comprising:

means for registering the function with the ARM agent prior to generation of the start and stop time markers.

24. (Previously Presented) The method of claim 22, further comprising means for causing generation of the start and stop time markers without modifying instructions associated with execution of the function.

25. (Previously Presented) The method of claim 22, further comprising means for a record

corresponding to the function for storing the response time associated with the function.

26. (Previously Presented) The method of claim 25, wherein the record includes a field for identifying a parent, if any, of the function in a hierarchical parent-child transaction chain.

27. (Previously Presented) The method of claim 26, wherein the record includes another field for identifying a top level transaction in the parent-child transaction chain.

28. (Previously Presented) The method of claim 22, wherein the software component includes at least one of the following: a server page, a servlet of a server side component, a driver, a naming and directory interface (NDI) or remote method invocation (RMI) component.

29. (Previously Presented) The method of claim 28, wherein the function of the software component includes at least one of the following: a service method of a server page, a doFilter, a doGet, a doPost or a service method of a servlet, a getConnection, executeQuery, or selected methods of driver, or remote, local or home interface methods of a server side component.

30. (New) The method of claim 1, further comprising generating at least one other correlator, the at least one other correlator corresponding to the at least one other function.